IN-14
43077

RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

p14

Final Report of the Telescience Workstation

Robert L. Brown
Dee Doyle
Richard Haines
Michael Slocum

RIACS
Research Institute for Advanced Computer Science
NASA Ames Research Center, 230-5
Moffett Field, CA 94035-4000

RIACS Technical Report 89.1

(NASA-CR-188878) TELESCIENCE WORKSTATION Final Report (Research Inst. for Advanced Computer Science) 14 p CSCL 14B

N92-11033

Unclas

G3/14 0043077

Final Report of the Telescience Workstation

Robert L. Brown
Dee Doyle
Richard Haines
Michael Slocum

RIACS
Research Institute for Advanced Computer Science
NASA Ames Research Center, 230-5
Moffett Field, CA 94035-4000

1 Abstract

As part of the Telescience Testbed Pilot Program USRA/RIACS proposed to support remote communication by providing a network of human/machine interfaces, computer resources, and experimental equipment which allows: remote science, collaboration, technical exchange and multimedia communication. The telescience workstation is intended to provide a local computing environment for telescience.

We proposed as part of the program:

1) To provide a suitable environment to integrate existing and new software for a telescience workstation.

2) To provide a suitable environment to develop new software in support of telescience activities.

3) To provide an interoperable environment so that a wide variety of workstations may be used in the telescience program.

4) To provide a supportive infrastructure and a common software base.

5) To advance, apply, and evaluate the telescience technology base.

We created and deployed a prototype telescience computing environment, designed to bring practicing scientists in domains other than computer science into a modern style of doing their computing. This environment, which we named the "Telescience Windowing Environment, Phase I", (TeleWEn-I), met some, but not all of the goals stated above. TeleWEn-I provided a window-based workstation environment and a set of tools for text

editing, document preparation, electronic mail, multimedia mail, raster manipulation, and system management.

2 Introduction

The concept behind telescience is that remote experiments and remote laboratories can be managed and manipulated from afar. Essential ingredients in any telescience implementation are the laboratory (which may be physically distributed), the computers (which may also be physically distributed) and the network interconnecting them.

Telescience capabilities offer more flexibility and opportunities for performing scientific experiments in environments that are man-hostile, such as outside Earth's atmosphere. NASA's initial interest in telescience is as a technology for controlling experiments onboard the Space Station in low Earth orbit. Hence, efforts are on-going to develop the technology base necessary for real and usable telescience applications.

The Telescience Workstation project intended to address those parts of the technology base that involve the computers and the software they run, asking the question, "What are the best models for computer interaction, for network abstractions, and for software development environments for telescience?" Our goal was to investigate these models and test our decisions by building a prototype implementation.

To accomplish our goals we provided an infrastructure support environment for workstations. This infrastructure included:

A network model. In our model, the network provides the scientist with access to remote laboratories and to remote resources, and is transparent to the scientist. Conducting an experiment at a remote laboratory is no different than conducting one in his own laboratory. The network brings the laboratory into the workstation. Each scientist has the full power and capabilities of all of NASA's laboratories at his fingertips. The network also brings all of the data in NASA's databases into the workstation for the scientist's perusal. The network allows the scientist to collaborate with colleagues on Earth and in space. Through the network, two or more scientist at disparate locations can jointly conduct experiments, view results, conduct research and write papers.

A workstation model. The workstation is the scientist's window into computing, providing access to more specialized machines, to laboratories, and to other resources. The workstation provides the mechanisms for manipulating the computing environment, visualizing experimental results, and for collaborating with other scientists. The workstation is an integral component of the scientist's environment assisting him in every task.

A cohesive underlying environment. A cohesive underlying environment provides for workstation interoperability. This allows scientists to share applications even though the underlying workstation architectures may differ. The environment provides for a common user interface. The steps to perform a task on one workstation, will also be the steps to performing the same task on another. This allows the scientist to work efficiently at any workstation. No longer will a scientist sit down at a new workstation and have no clues as to how to

Stay independent of any single vendor. An important consideration was to not become locked into any single workstation vendor. To achieve this, we needed to pick a development environment that would allow the creation of portable software.

Add value to the basic hardware and software. As they come from the vendors, most workstations and their associated software are generic; there are few tools that are oriented specifically towards telescience needs. We wanted, as a part of the TCE, to include some generally useful, yet telescience oriented, tools.

Provide a development environment tailored to telescience applications. Program development environments are usually not tailored to a particular computational domain, yet each domain often has special requirements. In telescience, we expect that almost every application will need to use one or more computer networks routinely for access to remote resources. Many applications will need to manipulate custom hardware that is tightly coupled to some machine, perhaps even the local workstation. Plus, portions of telescience applications will require advanced interactive user interfaces and graphics. We sought to identify components of a development environment that fulfilled these needs.

Provide an integration mechanism for telescience software. There is a single contraint on the environment package, namely that it be minimally intrusive when installed in the user's current computer system. This means that when the user installs this software it does not conflicts with any other software that may already be in place. We believe that from minimal intrusion comes further benefit such as ease of installation and maintainance. Achieving minimal intrusion seems like a very reasonable goal, and in the Testbed Pilot Program was mandatory. The installers potentially ranged from computer-naive scientists to Ph.D. computer researchers and computer system programmers. Many of these users have neither the time nor inclination to insure our software would be compatible with their system, or to delve into the depths of the software to retrieve something usable. Often an installation site may be composed of a single researcher who needs the tools of the environment, but cannot afford to hire a system programmer to support him/her. Even in a situation where the installer is a highly sophisticated computer user, we hypothesize that the principle of minimal intrusion is desirable, and will save the end user/installer much pain and grief. However, for this report we take the constraint as given.

Provide standards for development and documentation. Because the requirements of telescience applications are specialized, we sought to develop a standard for programming technique, but not programming language or style. The advantage of standards for development is that it increases the interoperability of applications and sharing of code; difficult tasks solved by one developer can then be used by another. Documentation standards are primarily quality and style standards, and we hoped to outline some guidelines.

4 The TeleWEn-I Prototype

We worked to produce a first-cut computing environment that would validate and meet the design goals stated above. Recognizing that we did not have sufficient time or funding to build the ultimate TCE, we chose to create and distribute a system that provided the telescience working community with a modern workstation environment. We initiated a hardment for integration and support currently as many of the needed tools are developed and available in Unix.

4.1.3 Evaluation of windowing environments

The window system is responsible for rendering primitive graphics objects and coordinating use of shared resources. Windowing systems come in two flavors, device-independent and kernel-based. Device-independent systems run on many platforms while presenting the user with a consistent interface. Kernel-based window systems are integrated into the native operating systems and have access to kernel level routines. The device independent windowing systems considered were NeWS, X and Andrew. The kernel-based system considered was SunView, provided bundled with SMI workstations

The X window system is developed at MIT has a client-server foundation. X uses a pixel-based imaging model.

NeWS is a Sun Microsystems window system having a client-server foundation. NeWS provides a stencil-paint imaging model of the PostScript language.

The Andrew window system is developed at CMU has a client-server foundation.

SunView is SMI kernel based window system. SunView was chosen because of its stability. The X Window System Version 10.4 was known to have many problems and the new and improved Version 11.1 was due soon. Neither NeWS nor Andrew was sufficiently well-known or mature for serious consideration.

We acknowledged that the selection of SunView violated the vendor-independence criterian, but decided the benefits of stability and maturity outweighed the defficiencies. In future TeleWEn implementations, we would select an X-based windowing system and ultimately a PostScript-based system.

4.1.4 Problems

Once workstation hardware and operating system have been selected, there are several problems associated with managing a large software environment composed of many separate packages of code, documentation, and data, which is to be distributed to other places. These problems are individually described below.

How to partition the software environment into a filesystem structure. In a sense, the filesystem structure of as large software environment is driven by the requirements of the software to be placed in that structure, and the manner in which that software will be used. This structure must not only make sense in terms of maintenance and enhancement by the developer(s), but must provide straight forward access to those who will be using the environment in the field. A good deal of thought must be placed on the desired end user filesystem structure, and how that may differ from the development structure.

The TeleWEn-I distribution meets the goal of minimal intrusion into a standing Unix filesystem, and attempts to follow the Unix naming conventions. The environment was develbroad categories of software, specifically, document preparation, electronic mail, time management, system management, communication management, and a good text editor. The specific software packages we selected are listed in an appendix.

How to distribute this software under the minimal intrusion constraint. First, make it simple to distribute whatever you have. If it takes more than one command to make a distribution, you are probably doing something wrong. That command script or make file which actually does the work may be extremely complicated, and ask lots of questions. But the effort will be paid off the first time you forget what all the steps that you needed to take to assemble everything, or the first time someone else needs to make a distribution.

Second, we have found that simplicity in the structure of the distribution is also important, given the range of end user sophistication. The mechanism we use to distribute the TeleWEn is a set of archive files created by the Unix "tar" program. Each file is constrained to be as close to 1 megabyte as possible to make these files more transportable over the Internet. The files are not compressed, to simplify installation at the user sites, especially those with space constraints.

We provide two distribution mechanisms, transfer via network file transfer (ftp) and physical tapes. Ftp is the preferred method, keeping in line with the philosophy of Telescience, which is to promote remote access. Thus, if the 'tar' files become much larger than 1 megabyte, then "ftp" will often fail to complete the transfer across the Internet. Make the files much smaller and the numbers of files start to climb beyond managability.

The binary and source tar files are kept in a directory, available for anonymous ftp across the network. These same files may be directly copied (with 'dd') onto tape(s) for distribution. The site installation script ascertains whether the installation is via tape or copied ftp files. However, the installation mechanism is the same, the component files are extracted from the tar file (on tape or disk).

As mentioned earlier, providing a distribution of minimal size for installation at sites with limited disk space is an important consideration. We have separated the binary and source tar files into separate directories. This allows anonymous ftpers to copy all of the binaries, or all of the sources, with a single ftp command.

For those sites with extremely limited space, it also possible to extract only subsets of the binary or source distribution. This is accomplished by retrieving the desired TAR files from tape or network. The installation script recognizes that missing files should be announced, but that the omission is not an error. However, it is up to a user at the site to insure that the subset is complete enough to function. We feel that those sites which deviate farthest from the standard installation, should pay the heavier price in terms of required skill and effort. By design, those that follow installation instructions exactly should, similarly, have no problems.

How to provide an upgrade path between versions of the software package. Since there are already versions of TeleWEn out, it would be nice to provide and upgrade path to

o 6_Telescience_System_Management_Guide
o 7_Telescience_Users_Guide
5. man pages for (almost all) executables
6. bibtex-documentation in dvi format

Packages
1. Rand MH mail handling package
2. GnuEmacs
3. TeX, plus public domain fonts
4. LaTeX
5. bibtex, a (La)TeX bibliography processor
6. slitex, a LaTeX viewgraph generator
7. s2latex, a scribe to LaTeX translator
8. DVI to Postscript filter, to convert (La)TeX output to that suitable for a postscript printer
9. TeX utilities
10. metafont (not installed as an executable)
11. Utah graphics tool kit
12. calen, Wisconsin calendar program
13. symbolic debugger (gdb), that comes with GnuEmacs
14. kermit
15. Macintosh communication programs macput and macget
16. rdist, a remote file distribution service
17. Frame Maker, a "What you see is what you get" text/picture editor
18. netup, a network/host monitoring program
19. generic user files, to set up telescience user accounts
20. GnuEmacs lisp "glue" code to enhance the emacs and MH user interfaces.

Etc Files
1. example.printcap
2. termcap

Scripts
1. installrelease, assists you in installing this software environment
2. copyuser, assists you in creating telescience user accounts

5.3 User Evaluations

After the TeleWEn package had been distributed for approximately two months, a RIACS human factors research scientist designed and administered a user survey to evaluate TeleWEn's effectiveness in the TTPP. This was done using electronic mail so that each respondent could reply with minimum effort and time. A major objective of this activity was to determine the types of uses they made of workstations and the extent they felt such workstations, running the TeleWEn package enhanced their productivity.

The survey had several objective which were divided into the Pre-TelWEn receival period (to assess who the TTPP participants had conducted their work before receiving the software package) and the Post-TeleWEn receival period to assess its impact. Questions

Table 1

Survey Results

Question	Respondent A	B	C	D
1. Occupational Category	PI	Administrator	Research	System Programmer
2. Prime Discipline	Earth Science	Technology	Astronomy	Computer Science
3. Activities Computer is used for:				
Access to other computers	X	X	X	X
Compiling	X		X	X
Conferencing		X		
Data Analysis	X		X	X
Data Collection	X	X		X
Data Generation	X	X	X	X
Data Viewing	X	X	X	X
Document Preparation	X	X		X
Electronic mail processing	X	X	X	X
Experiment Design			X	
Experiment Development			X	
Experiment Evaluation			X	
General Text Editing		X	X	X
Graphical Editing	X	X		
Hardware Evaluation	X			
Software Design			X	X
Software Development	X		X	X
Software Evaluation		X	X	X
Use 3rd Party Software	X	X		X
Use Vendor Utilities	X	X		X
Windowing	X	X	X	
Word Processing		X		
Total =	14	14	14	14

Table 1 (continued)

	A	B	C	D
13. Any direct impact in following areas: (cont'd)				
Ability to:				
(e) Influence overall labor time effectively	-----	9	-----	-----
(f) Influence overall workload	-----	9	-----	-----
(g) Influence programming/ development time	-----	N/O	-----	-----

14. List name of software used before receiving TeleWEn and rate it for each listed work element. Then rate TeleWEn's relative capabilities to perform the same work element. There were two responses received this question. See note 2 for the scoring key.

Work Element	(rating score)	(rating score)
(a) Data plotting/graphics	None	Internally dev. pkgs.(3)
(b) Document preparation	emacs/troff (blank)	troff(7)
(c) Electronic Mail	Mh (blank)	sendmail (7)
(d) General Editing	emacs (blank)	vi (8)
(e) Graphical Editing	None (blank)	None
(f) Network Monitoring	None (blank)	None
(g) Word Processing	emacs (blank)	None

18. What else would you like to see included in future releases of TeleWEn?

Respondent 3 answered:

"Suggested data compression algorithms, another pc-to-mainframe communication program (other than Kermit) that takes advantage of the higher speed asynchronous modems."

19. Any other comments?

Respondent 3 answered:

"Response to our request for the TeleWEn was prompt, and I apologize for not being as prompt in experimenting with some of the software. Our testbed program development priorities required that we install other packages first, especially in light of the disk storage limitations."

The following comments and observations refer to the data given in Table 1

Question 3 dealt with how each user actually used their computer(s) in their daily work. While each of the four respondents checked fourteen activities there was diversity which appeared to be related to their own occupational category in most cases. All four respondents indicated that they used their computer(s) to (a) access other computers, (b) perform data generation, (c) perform data viewing, and (d) perform electronic mail processing. Three respondents indicated that they used their computers to: (a) compile programs, (b) analyze data, (c) collect data, (d) prepare documents, (e) perform general text editing, (f) perform software evaluation, (g) use third party software, and (h) do windowing. In summary, of the 22 categories provided, at least one respondent checked every one suggesting that computers are being used in a very wide variety of ways. Of course it is problematic whether TeleWEn would be considered *the* software package of choice.

Question 4 related to what operating systems the respondent has used. All four had used UNIX, three VMS, two DOS, and one AOS (Data General), Apple- Max and HP.

Question 5 showed that TeleWEn was distributed relatively late in the TTPP activity. Nevertheless, some respondents indicated that the large amount of resident disk space required for TeleWEn prevented them from implementing it as early (or at all) as they would have liked to.

cult parts of programming.

The vendor-independent criterion was not satisfied by TeleWEn-I because of the selection of the SunView windowing system. Work is in progress to redesign the system to use X-windows, and to provide support for more types of workstations and operating systems.

TeleWEn-I was designed to provide a complete workstation "world" in which the workstation user lived. In order to use the tools provided, the user would have to absorb large pieces of the TeleWEn-I system into his own environment, including login files, window system configuration files, etc. This resulted in configuration control problems, such as when a user wanted to customize a portion of the environment, which would cause portions of TeleWEn to no longer work.

Many of the value-added features of TeleWEn-I were integrated into the text editor, GNU emacs. This proved to be a weakness, because emacs no longer behaved as the GNU documentation said it would. In the future, we will try to avoid integrating so much functionality into specific portions of the distribution.

merge .... Merges the differences between divergent decendant files.
splittar .... Splits a large tar file into a smaller set of tar files.
tick .... Generate output at a specified interval.
uniquekey .... Writes the system clock time to the standard output.
zcat .... Compress and expand data, see compress.
zcmp .... , zdiffCompare compressed files.
zmore .... File perusal filter for crt viewing of compressed text.
.cshrc .... An exmple of the first file read by the Unix shell at startup.
.login .... An example of the second file read by the Unix shell at startup.

.TeleCommunications - File Transfer

kermit .... File transfer program.
ftp .... File transfer program.
macget .... Receive file from Macintosh via modem7/MacTerminal.
macput .... Send file to Macintosh via modem7/MacTerminal.
rdist .... Remote file distribution service.

TeleCommunications - Mail

The Rand MH mail handling package was provided with the following features:

mhmail .... Send or receive mail.
next .... Shows the next mh mail message.
packf .... Compress an mh folder into a single file.
folder .... Set current mh mail folder.
folders .... List current mh mail folders.
forw .... Forward mail messages.
mark .... Mark mh mail messages.
inc .... Incorporate new mail into mh mailbox.
mhpath .... Print full pathnames of MH messages and folders.
burst .... Explode digests into messages.
dist .... Redistribute a message to additional addresses.
msgchk .... Check for mail messages.
msh .... MH shell (and BBoard reader).
pick .... Select mh mail messages by content.
prev .... Show the previous mh mail message.
refile .... File mh mail message in other folders.
repl .... Reply to a mh mail message.
rmf .... Remove mh mail folder.
rmm .... Remove mh mail messages.
show .... Show (list) mh mail messages.
scan .... Produce a one line per mh mail message scan listing.
send .... Send a mh mail message.
sortm .... Sort mh mail messages.
vmh .... Visual front-end to MH.
whatnow .... Prompting front-end for send.
whom .... Report to whom a mh mail message would go.
prompter .... Prompting editor front-end.
tarmail .... Encode binary to printable ASCII.
untarmail .... Decode binary to printable ASCII.

2882 Sand Hill Road
Suite #210
Menlo Park, CA 94025
Keith Lantz <lantz@orc.olivetti.com>

Richard Murphy
Southwest Research Institute
Division of Space Sciences
6220 Culebra Road
San Antonio, TX 78284
Richard Murphy <richard%swrimv@io.arc.nasa.gov>

Larry Schooley and Richard Bienz
Electrical & Computer Engineering Building
University of Arizona
Tucson, AZ 85721
Larry Schooley <schooley%evax2@jvax.ccit.arizona.edu>
Richard Bienz <bienz%looney.span@io.arc.nasa.gov>

George B. Adams
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907
George Adams <gba@ecn.purdue.edu>

Jim Bigham and Glyn LeBlanc
Nasa Johnson Space Center
Houston, TX 77058
Jim Bigham <jbigham%nasamail@ames.arc.nasa.gov>
Glyn LeBlanc <gleblanc%nasamail@ames.arc.nasa.gov>

Rock Bush
Electrical Engineering Department
Starlab/SEL
Stanford University
Stanford, CA 94305-4055
Rock Bush <bush@star.stanford.edu>

Supriya Chakrabarti and George Kaplan
Space Sciences Laboratory
University of California
Berkely, CA 94720
Supriya Chakrabarti <supc@sag2.ssl.berkeley.edu>
George Kaplan <gckaplan@sag2.ssl.berkeley.edu>

Terry Herter
Center for Radiophysics and Space Research

Cornell University
Ithaca, NY 14853-0352
Terry Herter <herter@crnlastr.bitnet>

David Koch and Charles Maxson
Harvard-Smithsonian Center for Astrophysics
60 Garden St., MS 65
Cambridge, MA 02318
Dave koch <koch@cfa.harvard.edu>
Chaarles Maxson <maxson@cfa.harvard.edu>

Karen Moe and Curtis Emerson
Mail Code 502
Goddard Space Flight Center
NASA
Greenbelt MD 20771
Karen Moe <kmoe%gsfcmail@ames.arpa>
Curtis Emerson <emerson@palantir.gsfc.nasa.gov>

Peter Shames
Space Telescope Science Institute
3700 San Martin Drive
Homewood Campus
Baltimore, MD 21218
Peter Shames <pmbs@stsci.arpa>